

1 INTRODUÇÃO

Um sistema embarcado pode ser definido como um dispositivo eletrônico microprocessado construído para executar uma ou mais funções, frequentemente interagindo com o mundo (HEATH, 2002). O termo ou jargão moderno “embarcado”, traduzido equivocadamente para a língua portuguesa a partir da palavra inglesa “*embedded*”, que em tradução correta quer dizer “embutido”, não informa em sua plenitude a abrangência do conceito, pois um sistema eletrônico pode estar “embarcado” sem estar “embutido”. O conceito de embutido envolve muito mais que o entendimento de embarcar. Envolve levar em conta no projeto aspectos e especificações relativas à ocupação de espaços físicos diminutos, consumo de energia, dissipação de calor, interferência eletromagnética, entre outros. Para citar exemplos, temos equipamentos modernos como aparelhos televisores, robôs de exploração espacial, fornos de micro-ondas, caixas eletrônicos, relógios de pulso, satélites de telecomunicações, impressoras, telefones celulares, Controladores Lógicos Programáveis (CLP) e brinquedos, abrangendo aplicações médicas, industriais, comerciais e de lazer. Segundo (MARKETSANDMARKETS, 2020), o mercado global de sistemas embarcados movimentou cerca de 87 bilhões de dólares em 2019 e estima-se que irá movimentar aproximadamente 119 bilhões de dólares em 2025, com um crescimento anual esperado de 5,36% ao ano. O ramo de sistemas embarcados tem crescido em importância e em complexidade, tornando-se também um campo estratégico para o desenvolvimento tecnológico de qualquer país.

Além disso, existe uma lacuna entre o que é ensinado em cursos de graduação que abordam sistemas embarcados, e os desafios impostos pelos projetos de desenvolvimento no mundo real. As habilidades adquiridas ao longo da vida acadêmica dos estudantes permitem amplo domínio de conhecimentos técnicos (teoria e prática), mas não preparam os estudantes, por exemplo, para comunicação com clientes ou interessados leigos, dimensionamento do problema real e da sua solução.

O objetivo deste artigo é a proposição de uma *abordagem baseada em problemas para o ensino de técnicas de desenvolvimento e projeto de sistemas eletrônicos embarcados (embutidos) visando estimular os estudantes e dar sentido prático aos conceitos apresentados, aproximando-os às situações que eles vivenciarão em problemas do mundo real. É feita a concepção de um roteiro de desenvolvimento para o exemplo de um módulo eletrônico com aplicações em robótica, visando o PBL (Project-Based Learning) no ensino de sistemas embarcados. Neste roteiro são definidas as etapas a serem seguidas, formas de abordar problemas e conceber soluções. O módulo eletrônico exemplo a ser desenvolvido, aqui denominado computador de locomoção, deverá equipar uma base robótica móvel chamada HiBot, que tem sido desenvolvido no Laboratório de Robótica do Departamento de Engenharia Elétrica da Universidade Federal da Bahia (UFBA). O HiBot tem como finalidade servir de plataforma para testes de algoritmos voltados para modelagem, controle, robótica e interações entre humanos e robôs.*

O restante deste artigo está organizado da seguinte forma. A seção 2 introduz uma metodologia para desenvolvimento de sistemas embarcados. A seção 3 mostra a estrutura básica de um sistema embarcado aplicado à robótica. Na seção 4 são apresentados alguns requisitos para o projeto do controlador de locomoção, enquanto que na seção 5 são levantadas arquiteturas e soluções tecnológicas visando atender a tais

requisitos. Na seção 6 é apresentado o módulo eletrônico projetado e resultados de testes. Por fim, a seção 7 traz as considerações finais.

2 METODOLOGIA DE DESENVOLVIMENTO DE SISTEMAS EMBARCADOS

A abordagem com a qual se deseja familiarizar os estudantes consiste em apresentar formalmente as seguintes etapas de desenvolvimento: levantamento de requisitos, especificação, elaboração da arquitetura, desenvolvimento dos componentes e integração (WOLF, 2001). A primeira etapa consiste em levantar requisitos, ou seja, obter descrições de clientes, consumidores e interessados sobre o que se deseja do sistema a nível de comportamento. As especificações são descrições mais detalhadas do que se deseja do sistema também a nível comportamental. Uma arquitetura é um conjunto de diagramas que detalham a estrutura do sistema em termos de seus componentes. Os componentes, no contexto de sistemas embarcados, são módulos de *hardware* e *software* que possuem funções e operações definidas, como por exemplo um circuito completo de um *driver* para motor elétrico ou um *device driver* para interface com um acelerômetro. Por fim, é realizada a união dos componentes para a obtenção do sistema, chamada de integração.

3 SISTEMAS EMBARCADOS E BASES ROBÓTICAS MÓVEIS

Segundo (ISO10218-1, 2011), um robô é definido como "*uma máquina manipuladora, com vários graus de liberdade, controlada automaticamente, reprogramável, multifuncional, que pode ter base fixa ou móvel para utilização em aplicações de automação industrial*". A classe dos robôs móveis terrestres tem se destacado pela sua aplicabilidade em diversas tarefas, incluindo exploração espacial, limpeza doméstica, transporte autônomo de cargas, exploração oceânica, apoio emocional e inspeção em dutos. Segundo (SIEGWART, 2004), os problemas básicos em robótica móvel consistem em locomoção, percepção, cognição e navegação. O problema da locomoção é resolvido através do entendimento sobre cinemática, dinâmica e teoria de controle. Percepção envolve análises de sinais e áreas especializadas como computação visual e tecnologias de sensores. A cognição é responsável por ações e comportamentos exibidos com base na leitura de sensores e nos objetivos estabelecidos. A navegação requer conhecimentos sobre algoritmos de planejamento, teoria da informação e inteligência artificial (RUBIO, 2019).

Existe uma estrutura bastante comum para a solução dos problemas de locomoção e navegação, que consiste em utilizar um computador (comumente disponível nas mãos dos desenvolvedores) ou uma SBC (*Single Board Computer*), capaz de executar um sistema operacional de alto nível como o Linux e um *framework* de robótica como o ROS (*Robot Operating System*), para a execução das funções de navegação, enquanto um ou mais microcontroladores de pequena ou média escala ficam responsáveis pela locomoção e outros detalhes de operação considerados de baixo nível. O computador de navegação pode, inclusive, ser instalado em uma base estacionária, fora do robô, e se comunicar com o computador de locomoção através de uma rede sem fio qualquer. Este trabalho concentrar-se-á em desenvolver um computador de locomoção.

O desenvolvimento de um computador de locomoção para auxílio no ensino do desenvolvimento de sistemas embarcados se justifica pois ele trás aspectos importantes como seleção de interface com sensores e atuadores (motores), acionamento e controle de atuadores, implementação de protocolos de rede.

4 REQUISITOS

O ponto de partida para a concepção do computador de locomoção é o levantamento das necessidades daqueles que irão utilizá-lo. Esta informação pode ser obtida através de pesquisa por trabalhos publicados, entrevista estruturada e levantamento de sensores e atuadores de alguma base móvel pré-existente (tecnologia prévia, ação também conhecida como análise competitiva).

Como resultado de pesquisas e entrevistas, foi verificado que o sistema deveria ser capaz de executar o seguinte: (i) controle de tração para as rodas, (ii) controle de velocidade, (iii) modelo cinemático, (iv) fusão de dados de sensores e filtros de Kalman, e (v) controle de trajetória com realimentação de posição cartesiana.

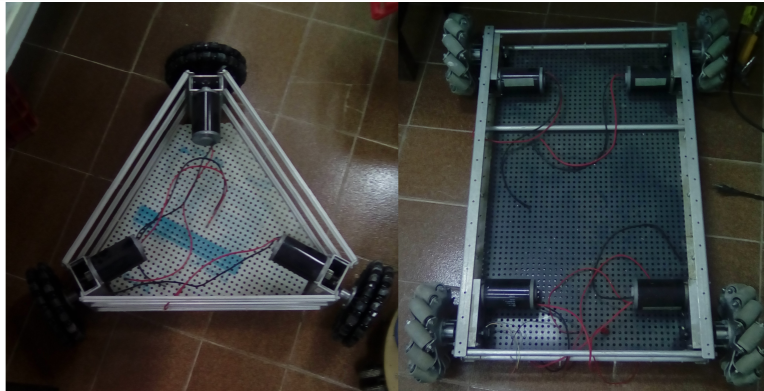
A partir do levantamento de atuadores e sensores (Figura 1) disponibilizados previamente para a base móvel do HiBot (Figura 2), foram extraídos os requisitos adicionais: tensão de alimentação de 12 V fornecida por baterias; acionamento de bases com até quatro rodas; leitura de correntes até 30 A nos terminais dos motores para controle de tração; interface para *encoders* rotativos com contagem elevada de pulsos por volta.

Figura 1 – Motor modelo PM25R-45F-1003 e *encoder* modelo EP4-250-250-N-S-D-D-B.



Fonte: andymark.com.

Figura 2 – Bases robóticas móveis omnidirecionais de três e quatro rodas.



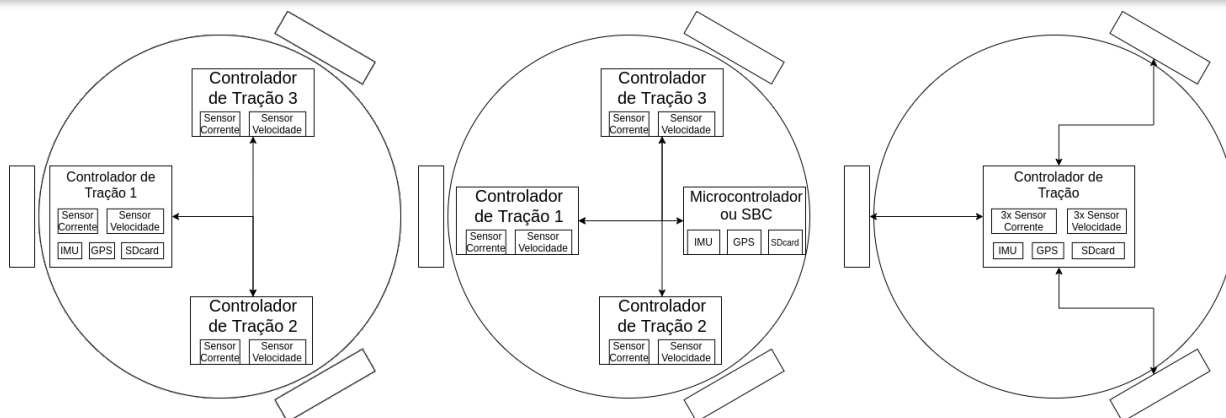
Fonte: Autores.

Detalhes levantados durante entrevistas revelaram alguns aspectos relacionados à operação, onde deve-se: permitir configuração *on-line* de parâmetros, incluindo-se modos de operação e ganhos de controladores; permitir telemetria e teleoperação; possuir canal para comunicação com SBCs, por onde o computador de locomoção possa ser configurado, monitorado e controlado; permitir que seu programador também consiga utilizá-lo para acionar a base móvel de forma isolada, *standalone*, sem depender da SBC.

5 CONCEPÇÃO DO MÓDULO ELETRÔNICO

A arquitetura de *hardware* consiste na distribuição das unidades de processamento, sensores, atuadores e comunicação em um sistema embarcado. Neste contexto, o computador de locomoção pode assumir ao menos três configurações distintas. A primeira envolve a divisão do computador de locomoção em dispositivos conectados em rede, cada um responsável pelo acionamento e controle de um motor, enquanto um desses opera também como centralizador de informações e realiza funções como telemetria, teleoperação e configuração remota. A segunda configuração simplifica o projeto dos controladores através da adição de um novo módulo para assumir as funções do centralizador de informações. Por fim, uma configuração onde todas as funções encontram-se centralizadas em um único módulo. As vantagens desta última configuração em relação às outras é a quantidade reduzida de microcontroladores e outros elementos no projeto, reduzindo custo e número de *firmwares* diferentes por base robótica. Pode-se citar também a eliminação do protocolo para comunicação entre controladores, o que diminui a latência. Porém, a implementação de todos os requisitos recai sobre apenas um elemento que precisa ser maior e mais complexo, tanto em *hardware* quanto em *firmware*. Todas as configurações estão representadas na Figura 3.

Figura 3 – Arquiteturas de *hardware* para o computador de locomoção. Da esquerda para a direita: modelo distribuído 1, modelo distribuído 2 e modelo centralizado.



Fonte: Autores.

5.1 SENSORES E ATUADORES

A técnica mais comum para acionamento de motores de corrente contínua com inversão de velocidade é o uso de um circuito chamado ponte H, o qual é acionado com sinais PWM (*Pulse Width Modulated*). Os pontos a serem observados durante o dimensionamento do circuito são os valores das correntes que serão controladas em condições normais, em sobrecarga e durante a partida. Esta última costuma ser entre 5 e 10 vezes maior que a corrente nominal em um motor e deve ser evitada através de um circuito dedicado que detecte sobrecorrentes ou através das ações do controlador digital.

Dois tipos de sensores são necessários para locomoção com controle de tração: um para medir a velocidade e outro para a tração nas rodas. Medir a velocidade é uma tarefa simples, bastando conectar *encoders* rotativos em eixos nos motores ou nas caixas de redução. A escolha do princípio de operação e da interface elétrica está atada à aplicação. Os *encoders* que equipam as bases móveis são do tipo óptico incremental com saída digital em quadratura.

Quanto à tração, esta pode ser estimada a partir da corrente elétrica medida nos terminais do motor. Medição de corrente em motores é um assunto abrangente que engloba diversos tipos de sensores, métodos de derivação do sinal e características que dependem de cada técnica. Ademais, a sincronia entre o sinal PWM e o momento da aquisição do sinal de corrente é desejada visando melhorar a qualidade do sinal após a digitalização.

Sensores adicionais são necessários para que o computador consiga executar as operações em *firmware* são considerados requisitos. A realização do controle cartesiano de trajetória exige um sensor que forneça de forma direta a posição da base móvel, como o GPS (*Global Positioning System*), mas que funcione também em ambientes internos. Em adição, um sensor que permita medir a orientação também é desejável como fonte de informação para filtros de Kalman.

5.2 UNIDADE DE PROCESSAMENTO

A busca por um microcontrolador está atrelada aos tipos, quantidades e modos de operação dos periféricos necessários para a aplicação. Por outro lado, é importante ter

também uma noção sobre o custo computacional, tamanho do programa e uso de memória RAM (*Random Access Memory*). É possível realizar previsões baseadas no número de operações matemáticas, simulações e até mesmo a partir de análise e testes com dispositivos similares. Nos casos em que o grau de incerteza é alto, costuma-se desenvolver usando dispositivos superdimensionados, sob a pena do maior custo, para posterior redução das características visando adequar-se melhor à aplicação. Com estes tópicos em vista, a Tabela 1 traz os periféricos e as quantidades necessárias, a Tabela 2 traz uma estimativa para os tempos de amostragem para os sensores e ativação para os algoritmos e a Tabela 3 lista alguns microcontroladores de diferentes fabricantes.

Tabela 1 – Periféricos necessários para as interfaces de *hardware* para a arquitetura centralizada.

Dispositivo	Interface	Quantidade
Sensor de corrente	ADC	4
Encoder	QDC	4
Ponte H	PWM	8
IMU	IIC	1
Cartão microSD	SPI	1
GPS	UART	1
Comm SBC	Ethernet	1
Comm remota	UART	1

Fonte: Autores.

Tabela 2 – Lista não exaustiva de dados e parâmetros configuráveis e suas taxas de amostragem.

Parâmetro ou variável	Período	Quantidade no período	Direção
Corrente dos motores	0,5ms	4	Tx
Rotação dos motores	10ms	4	Tx
Aceleração linear	10ms	3	Tx
Velocidade angular	10ms	3	Tx
Campo magnético	10ms	3	Tx
Ações de controle de tração	1ms	4	Tx
Ganhos dos control. de tração	-	12	Tx/Rx
Ações de controle de velocidade	10ms	4	Tx
Setpoint de velocidade	10ms	4	Tx
Ganhos dos control. De velocidade	-	12	Tx/Rx
Roll, pitch, yaw	10ms	3	Tx
GPS	100ms	3	Tx
Ganhos dos control. de posição	100ms	12	Tx/Rx
Ações de controle de posição	100ms	3	Tx

Fonte: Autores.

Tabela 3 – Lista não exaustiva de microcontroladores e seus periféricos para o computador de locomoção.

Microcontrolador	Tipo (bits)	Freq. (MHz)	RAM (KB)	Flash (KB)	ADC	PWM	QDC	IIC	SPI	UART
ATMEGA328P	8	20	2	32	6	6	0	1	1	1
ATMEGA2560	8	16	8	256	16	16	0	1	1	4
DSPIC33EP256MC502	16	120	32	256	6	6	1	2	2	2
PIC32MK1024MCF100	32	120	256	1024	42	12	6	6	6	6
STM32F103C8T6	32	72	20	64	10	20	4	2	2	3

STM36F407VGT6	32	168	192	1024	16	41	6	3	3	6
---------------	----	-----	-----	------	----	----	---	---	---	---

Fonte: Autores.

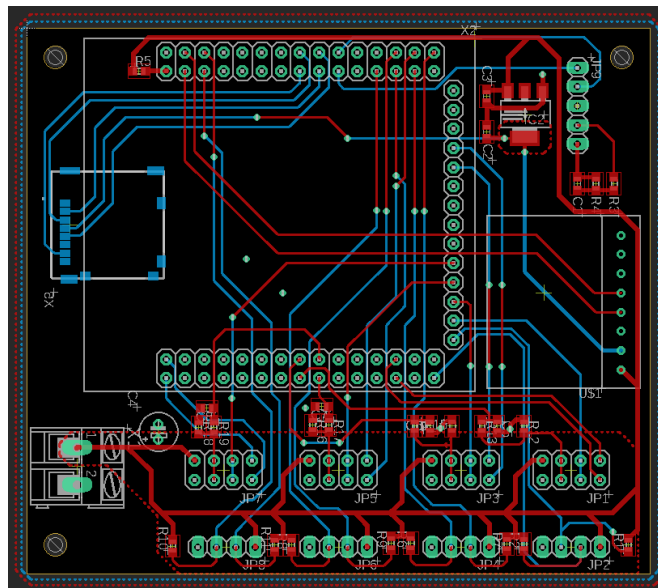
É importante notar que a interface para decodificação de sinais em quadratura (QDC – *Quadrature Decoder and Counter*) pode ser programada através de terminais de entrada e saída de uso geral com capacidade para gerar interrupções no microcontrolador. A vantagem desta abordagem é a maior flexibilidade no roteamento de terminais e suas funções, bom como na escolha do microcontrolador. No entanto, o alto número de interrupções realizadas na execução das funções principais pode degradar o desempenho do sistema.

No que diz respeito ao *firmware*, a aplicação deve ser tratada como sendo de tempo real, dadas as restrições temporais da Tabela 2. Aplicações de tempo real são aquelas que necessitam atender a requisitos temporais, ou seja, as tarefas da aplicação precisam ser executadas dentro de um período de tempo definido. As políticas de escalonamento mais básicas são a executiva cíclica (*cyclic executive*) e a de taxa monotônica (*rate-monotonic scheduling*). Na primeira, as tarefas são executadas cooperativamente, ou seja, uma segunda tarefa só pode assumir o controle do processador no momento em que uma primeira tarefa libere para uso o processador e/ou seus periféricos. Em alguns casos, para que a aplicação funcione como esperado, as tarefas podem fazer uso de máquinas de estado e interrupções para obter comportamento não bloqueante, também chamado de assíncrono, onde a operação desejada não é necessariamente finalizada na primeira vez em que uma função é chamada, sendo necessária a implementação de códigos de erro que informem o andamento da operação. Esse comportamento pode ser implementado, por exemplo, quando se aguarda a recepção ou a transmissão de um pacote de rede ou o fim da operação de um dispositivo externo. Na segunda, o sistema é capaz de interromper a execução de uma tarefa de menor prioridade para executar uma tarefa desejada com maior prioridade, realizando um uso mais eficaz do tempo de processador, em alguns casos, e com maior garantia do atendimento dos requisitos temporais de tarefas críticas. O *freeRTOS*, que é um sistema operacional de tempo real amplamente utilizado por ser gratuito, implementa uma política de escalonamento mista entre a executiva cíclica e a de taxa monotônica.

6 DESENVOLVIMENTO DO PROTÓTIPO

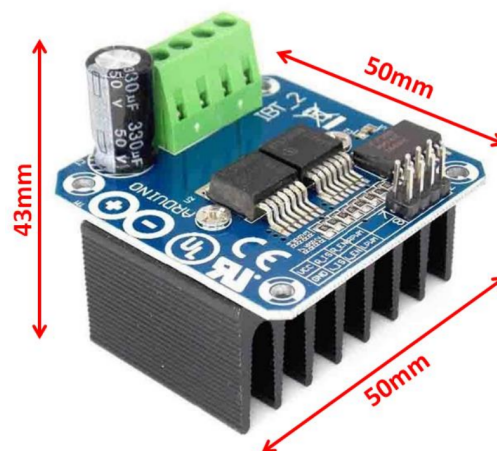
O computador de locomoção foi concebido como duas placas de circuito impresso empilháveis. A placa principal, Figura 4, contém o sensor de orientação e pontos de conexão com dispositivos instalados remotamente, como pontes H, sensores de corrente e *encoders*, que são instalados próximos dos motores. A placa secundária contém o microcontrolador e os componentes considerados básicos para a sua operação, como pontos de conexão para um gravador, reguladores de tensão e *jumpers*. Com esta configuração, pode-se utilizar qualquer microcontrolador na placa secundária desde que se respeite o padrão de conexão com a placa primária, bem como facilita a manutenção pois em caso de falhas capitais em alguma das placas é necessário apenas a substituição da placa danificada. Para o acionamento dos motores, optou-se pelo modelo BTS7960, que possui terminais para leitura de corrente e para o qual existem módulos, como os da Figura 5, à venda. A solução escolhida para medir a orientação consiste no acelerômetro e giroscópio de 3 eixos MPU6050 e no magnetômetro QMC5883L de 3 eixos. Ambos os sensores podem ser encontrados facilmente no módulo GY-87 e em seus clones, Figura 6.

Figura 4 – Computador de locomoção em ambiente CAD.



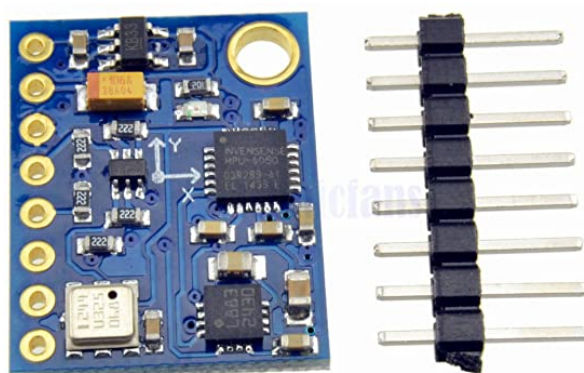
Fonte: Autores.

Figura 5 – Módulo pont H com BTS7960.



Fonte: handsontec.com

Figura 6 – Módulo IMU (Inertial Measurement Unit) GY-87.





Fonte: amazon.com

7 CONSIDERAÇÕES FINAIS

Neste artigo foi proposta uma abordagem baseada em problemas para o ensino de técnicas de desenvolvimento e projeto de sistemas embarcados. A aplicação da metodologia foi realizada através do exemplo do desenvolvimento de um módulo eletrônico com aplicações em robótica.

Um módulo eletrônico desenvolvido como exemplo, aqui denominado computador de locomoção, foi projetado para o acionamento de bases robóticas móveis com até quatro rodas. A arquitetura de *hardware* segue um padrão centralizado onde um microcontrolador é responsável pelas funções de telemetria, teleoperação, acionamento e controle.

Para estudos futuros sugere-se a aplicação da metodologia em sala de aula com posterior avaliação qualitativa e quantitativa do desempenho de alunos de engenharia.

Agradecimentos

O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

REFERÊNCIAS

HEATH, Steve. **Embedded Systems Design**. 2. ed, Burlington: Elsevier, 2002.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 10218-1: Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots**. Geneva, 2011.

MARKETSANDMARKETS, **Embedded System Market by Hardware, Software, System Size, Functionality, Application, Region – Global Forecast ro 2025**. Pune, 2020.

RUBIO, F.; VALERO, F.; LLOPIS-ALBERT, C. **A review of mobile robots: Concepts, methods, theoretical framework, and applications**. *International Journal of Advanced Robotic Systems*, v.16, n.2, 2019.

SIEGWART, Roland; NOURBAKHSH, Illah. **Introduction to autonomous mobile robots**. 1. ed, Cambridge: The MIT Press, 2004.

WOLF, Wayne. **Computers as components – principles of embedded computing system design**. 1. ed, San Francisco: Morgan Kaufmann Publishers, 2001.

**INSTRUCTIONS FOR PREPARATION AND SUBMISSION OF WORKS TO THE
SCIENTIFIC COMMITTEE OF XLIX BRAZILIAN CONGRESS OF ENGINEERING
EDUCATION – COBENGE 2021**



Abstract: *This article refers to the use of the problem-based approach for teaching development techniques and design of embedded electronic systems. To this end, the problem of designing and developing an electronic module for speed and traction control on an omnidirectional mobile robotic base is used as a way of introducing problem solving methods in the development of embedded systems. The use of the problem-based approach aims to encourage students to practice the basic concepts. The electronic module developed as an example, here called the locomotion computer, was designed to drive mobile robotic bases with up to four wheels. The hardware architecture follows a centralized standard where a microcontroller is responsible for the functions of telemetry, teleoperation, actuation and control.*

Keywords: *Problem-based learning. Traction control. Embedded systems. Mobile robotics.*